JP6202945

Publication Title:

SMALL ADDRESS- BASED MEMORY ACCESS METHOD AND COMPUTER SYSTEM

Abstract:

Abstract of JP 6202945

(A) PURPOSE: To enable a host (hypervisor) to perform access to an arbitrary storage location in an arbitrary guest zone in a large-sized memory when the host and a guest (OS) have small addresses. CONSTITUTION: A specific number of windows used by a host is provided. Each CPU in a system has one or a plurality of window address registers(WAR) 545 and one or a plurality of window registers 546-547. The host designates each page frame(PF) used as a host window in a host zone by using a WAR load instruction and each PF is correlated with the corresponding window number. When the host executes an arbitrary instruction having a small address by accessing the host window PF related to the said WR, the small address of a host operand is automatically replaced with a guest large address in the WR.

---

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平6－202945

(43)公開日　平成６年(1994)７月22日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　12/06	５６０　Ｂ	9366－5B		

審査請求　有　　請求項の数15　(全 21 頁)

(21)出願番号　特願平5－275633

(22)出願日　平成５年(1993)11月４日

(31)優先権主張番号　９７４３９３

(32)優先日　1992年11月10日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）

(72)発明者　ジョネル・ジョージ
アメリカ合衆国12569、ニューヨーク州プレザント・バレー、プラトー・ロード、アール・アール　６、ボックス　12

(74)代理人　弁理士　合田　潔　（外３名）

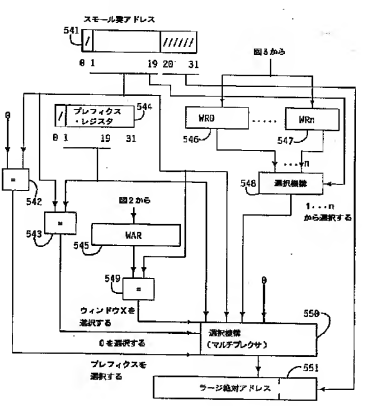
最終頁に続く

(54)【発明の名称】　スモール・アドレスを使用するメモリ・アクセス方法およびコンピュータ・システム

(57)【要約】

【目的】　ホスト（ハイパーバイザ）およびゲスト（ＯＳ）がスモール・アドレスを有する場合に、当該ホストが大型メモリ内の任意のゲスト・ゾーンの任意の記憶位置にアクセスできるようにする。

【構成】　ホストが使用する特定数のウィンドウが提供される。システム内の各ＣＰＵは、１つまたは複数のウィンドウ・アクセス・レジスタ（ＷＡＲ）と、１つまたは複数のウィンドウ・レジスタ（ＷＲ）を有する。ホストは、ＷＡＲロード命令を使用して、ホスト・ウィンドウとして使用される各ページ・フレーム（ＰＦ）をホスト・ゾーン内で指定し、各ＰＦはそれぞれのウィンドウ番号と関連付けられる。ホストが、当該ＷＲと関連するホスト・ウィンドウＰＦにアクセスして、スモール・アドレスを有する任意の命令を実行すると、自動的にホスト・オペランドのスモール・アドレスがＷＲ内のゲスト・ラージ・アドレスで置き換えられる。



1

【特許請求の範囲】

【請求項１】１つまたは複数のＣＰＵおよび大型メモリを具備し、１つまたは複数のゲスト・システム（ゲスト）およびホスト・システム（ホスト）がそれぞれのローカルなメモリ・ゾーン以外の記憶位置にアクセスできないスモール・アドレスを有する命令オペランドを使用するコンピュータ・システムにおいて、該ホストが、大型メモリ内の任意のゲスト・ゾーンの任意の記憶位置にアクセスするための方法であって、

上記ＣＰＵが、１つまたは複数のウィンドウ・アクセス・レジスタ（ＷＡＲ）と、該ＣＰＵ用に提供された１つまたは複数の指定ウィンドウにそれぞれが関連する１つまたは複数のウィンドウ・レジスタ（ＷＲ）とを含み、

ホスト・メモリ・ゾーン内の１つまたは複数のページ・フレーム（ＰＦ）を位置指定し、該位置指定された各ＰＦを、ＣＰＵの指定ウィンドウと関連付けてホスト・ウィンドウＰＦにするためのＷＡＲロード命令を実行するステップと、

ゲスト識別子およびゲスト・スモール・アドレスによって示されるゲスト・ゾーン内のゲスト位置を、指定ウィンドウおよび該指定ウィンドウに関連付けられたＷＲと関連付けるためのウィンドウ活動化命令を実行するステップと、

ホスト・オペランドが指定ウィンドウと関連付けられたホスト・ウィンドウＰＦをアドレスするとき上記ホストがゲスト位置にアクセスするようにする上記ゲスト識別子およびゲスト・スモール・アドレスから大型メモリ用のラージ絶対アドレスを生成し、上記指定ウィンドウと関連付けられたＷＲに該ラージ絶対アドレスをロードするステップとを含むメモリ・アクセス方法。

【請求項２】ホスト・ウィンドウＰＦをアドレス指定するオペランド・スモール・アドレスを有するホスト命令を実行するステップと、

ホストが、上記オペランド・スモール・アドレスによって、ラージ絶対アドレスで位置指定されるゲスト位置にアクセスできるようにするために、ＣＰＵのハードウェア／マイクロコードによって、該オペランド・スモール・アドレスを上記ＷＲ内のラージ絶対アドレスで置き換えるステップとを含む、請求項１に記載のメモリ・アクセス方法。

【請求項３】ホスト・ウィンドウＰＦ以外のホスト・ゾーン内のＰＦをアドレス指定するオペランド・スモール・アドレスを有するホスト命令を実行するステップと、

ＣＰＵのハードウェア／マイクロコードによって、上記オペランド・スモール・アドレスでアドレス指定される上記ホスト・ゾーン内の記憶位置にアクセスするステップとを含む、請求項１に記載のメモリ・アクセス方法。

【請求項４】それぞれが、ゾーン識別子用フィールドと、原点ラージ・アドレス用フィールドと、ゲスト・ゾーンの境界を定義するための限界ラージ・アドレス用の

2

フィールドとを有する複数のエントリを具備し、大型メモリ内の各ゲスト・ゾーンを位置指定するゾーン情報テーブル（ＺＩＴ）をメモリ内で提供するステップと、

上記限界ラージ・アドレスを、上記生成済みラージ絶対アドレスと比較し、該限界ラージ・アドレスを上回る生成済みラージ・アドレスを拒絶して、ホストが無効なゲスト・アドレスを使用するのを防止するステップとを含む、請求項１に記載のメモリ・アクセス方法。

【請求項５】それぞれが、ゾーン番号フィールドと、ゲスト・スモール・アドレス・フィールドと、ＷＲに現在関連しているゲスト位置を示すためのラージ絶対アドレス・フィールドとを含むとともに、ＣＰＵ内の指定ウィンドウおよび関連ＷＲとそれぞれが関連付けられた複数のエントリを具備するＣＰＵウィンドウ指定テーブル（ＷＳＴ）を提供するステップを含む、請求項１に記載のメモリ・アクセス方法。

【請求項６】上記ウィンドウ活動化命令を実行するとき、指定ウィンドウ用のゲスト・ゾーン識別子およびゲスト・スモール・アドレスを上記ＷＳＴのエントリに書き込むステップと、

生成されたゲスト・ラージ絶対アドレスを、上記ゲスト・スモール・アドレスが書き込まれた同じＷＳＴエントリに書き込むステップを含む、請求項５に記載のメモリ・アクセス方法。

【請求項７】ホスト・ウィンドウＰＦとして指定されるホスト・ゾーン内の１つまたは複数の連続するＰＦを位置指定するためのアドレスを、ＣＰＵの汎用レジスタにロードするステップと、

汎用レジスタ中のアドレスをＷＡＲにロードするＷＡＲロード命令を実行して、アドレス指定されたＰＦを、ホストが使用できるホスト・ウィンドウＰＦとして指定するステップとを含む、請求項１に記載の、メモリ・アクセス方法。

【請求項８】ホストによってアクセスされるゲスト・ゾーン内のゲスト位置についての、ウィンドウ識別子、スモール・アドレス、およびゲスト・ゾーン識別子を１つまたは複数のＣＰＵ汎用レジスタにロードするステップと、

汎用レジスタの内容を使用してウィンドウ活動化命令を実行し、ラージ絶対アドレスを生成すると共に、該汎用レジスタ中のウィンドウ番号に関連するＷＲにラージ絶対アドレスをロードするステップを含む、請求項４に記載のメモリ・アクセス方法。

【請求項９】ゲスト・ゾーン用のＺＩＴエントリ中の原点ラージ・アドレスに汎用レジスタ中のゲスト・スモール・アドレスを追加することによって、ラージ絶対アドレスを生成するステップを含む、請求項８に記載のメモリ・アクセス方法。

【請求項１０】ゲストがゲスト命令を実行している間にゲスト動作をインタセプトするステップと、

3

ホスト・テーブル中のゲスト状態記述子エントリに、インタセプトされたゲスト命令の記憶位置であるスモール・アドレスを取り込み、当該エントリのゲスト・ゾーン識別子を示すステップと、

ホストの指定ウィンドウ用のウィンドウ活動化命令を実行して、ゲスト・ゾーン中のインタセプトされたスモール・アドレス用のラージ・アドレスを生成し、該指定ウィンドウに関連するＷＲにそのラージ・アドレスを格納するステップと、

指定ウィンドウと関連するホスト・ウィンドウＰＦをアドレス指定するホスト・オペランド・アドレスを有するホスト命令を実行するとき、ホストがゲスト命令にアクセスして、ゲスト・ゾーンにゲスト命令を取り込むために、ホスト・オペランド・アドレスをＷＲ中のラージ・アドレスで置き換えるステップを含む、

請求項１に記載のメモリ・アクセス方法。

【請求項１１】ゲスト・スモール・アドレスを、プレフィクス・レジスタ中のプレフィクス・アドレスおよびアドレス・ゼロと同時に比較して、ゲスト・スモール・アドレスがプレフィクス・アドレスであるか、または逆プレフィクス・アドレスであるかを判定するステップと、

プレフィクス・レジスタまたはアドレス・ゼロに関してcompare equal条件が得られた場合に、ＷＲからラージ・アドレスを出力する代りに、プレフィクス・レジスタからプレフィクス・アドレスを出力し、またはアドレス・ゼロを逆プレフィクス・ラージ・アドレスとして出力するステップを含む、請求項１または１０に記載の、ホストが、大型メモリ内の任意のゲスト・ゾーンの任意の位置にアクセスできるようにする方法。

【請求項１２】単一のＷＡＲを使用して、ホストに対して指定されたウィンドウと関連する単一のホスト・ウィンドウＰＦを位置指定するステップ、または複数のＷＡＲを使用して、それぞれのＷＡＲが、ホストに対して指定されたそれぞれのウィンドウと関連する単一のホスト・ウィンドウＰＦを位置指定するステップ、または単一のＷＡＲを使用して、それぞれがホストに対して指定されたウィンドウと関連する複数のホスト・ウィンドウＰＦを位置指定するステップのいずれか１を含む、請求項１に記載のメモリ・アクセス方法。

【請求項１３】プログラムによってアクセス可能でないシステム・エリア記憶域内の活動ＺＩＴ中の対応するエントリと異なるエントリを有する置換ＺＩＴを、大型メモリ内のプログラム・アクセス可能エリアに書き込み、

活動ＺＩＴだけでシステムを制御するステップと、

システム内のすべてのＣＰＵによる実行を中断するステップと、

すべてのＣＰＵの実行が中断されている間に、システム内の選択されたプロセッサによって特殊コマンドを発行して、活動ＺＩＴを置換ＺＩＴでオーバーレイして、置換ＺＩＴを活動化するステップと、

4

各エントリに含まれるゲスト・スモール・アドレスとゾーン識別子を使用し、新たに活動化されたＺＩＴ中の識別ゾーン用のエントリを使用することによって、ＣＰＵウィンドウ指定テーブル中の各エントリに含まれるラージ・アドレスを再計算し、ＣＰＵウィンドウ指定テーブル・エントリおよび関連ＷＲに、再計算されたラージ・アドレスを再ロードするステップと、

ＣＰＵが新たに活動化されたＺＩＴ中でゾーンを使用できるように、すべてのＣＰＵの命令実行を再開して、システム内でのゾーンの再構成を完了するステップとを含む、システムを動的に再構成できる際に使用する、請求項４に記載のメモリ・アクセス方法。

【請求項１４】１つまたは複数のＣＰＵおよび大型メモリを具備し、１つまたは複数のゲスト・システム（ゲスト）およびホスト・システム（ホスト）が、メモリ内のそれぞれのローカルなメモリ・ゾーン以外の記憶位置にはアクセスできないスモール・アドレスを命令オペランドとして含むコンピュータ・システムにおいて、

ホスト・メモリ・ゾーン内の１つまたは複数のページ・フレーム（ＰＦ）を位置指定するためのホスト・スモール・アドレスを含む、１つまたは複数のウィンドウ・アクセス・レジスタ（ＷＡＲ）と、

それぞれが、１つまたは複数のホスト・ウィンドウに関連付けられ、ゲスト位置をアクセスするためのラージ絶対アドレスを含む、１つまたは複数のウィンドウ・レジスタ（ＷＲ）と、

ＷＡＲロード命令に応答して、上記ＰＦを上記ウィンドウと関連付ける手段と、

ウィンドウ活動化命令に応答して、ゲスト識別子およびゲスト・スモール・アドレスによって示されるゲスト・ゾーン内のゲスト位置を、上記ウィンドウおよびＷＲと関連付ける手段と、

上記ゲスト識別子およびゲスト・スモール・アドレスの内容に基づいて、上記ラージ絶対アドレスを生成し、該ラージ絶対アドレスを上記ＷＲにロードする手段とを有する、コンピュータ・システム。

【請求項１５】それぞれが、ゾーン識別子用フィールドと、原点ラージ・アドレス用フィールドと、ゲスト・ゾーンの境界を定義するための限界ラージ・アドレス用フィールドとを有する複数のエントリを具備し、大型メモリ内の各ゲスト・ゾーンを位置指定するゾーン情報テーブル（ＺＩＴ）を有する、請求項１４に記載のコンピュータ・システム。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、ホストとそのゲストが、それぞれのローカル・ゾーン内の記憶位置にしかアドレスできないスモール・アドレスを使用中のときに、ホストが大型メモリにある任意のゲスト・ゾーンの任意の記憶位置にアクセスできるようにするものである。

5

【0002】

【従来の技術】本明細書で「スモール・アドレス（小型もしくは小規模アドレスともいう）」とは、16ビット、20ビット、24ビット、31ビット、または32ビットのいずれかの一般アドレス・サイズを使用するものと定義する。本明細書で「ラージ・アドレス（大型もしくは大規模アドレスともいう）」とは、使用中のスモール・アドレスよりも大きなサイズを持つものと定義する。多数の既存のプログラムがスモール・サイズの仮想アドレスを使用しており、それらのアドレスは、スモール実アドレスまたはスモール絶対アドレスに変換される。変換された実アドレスまたは絶対アドレスが、メモリにアクセスするのに使用される。

【0003】従来の設計のホスト（ハイパバイザ）においては、ホストが使用するサイズよりも大きなアドレス・サイズを必要とする大型メモリ内のどの位置にもアクセスすることができなかった。従来型のホストおよびそのゲスト（オペレーティング・システム、OS）は、スモール・サイズ・アドレス（たとえば、最大2^{31}バイト・サイズ・メモリしかアドレスできない31ビット・アドレス）を使用するように設計されていた。OSはそれぞれ、コンピュータ・システムのメモリ内の異なるゾーンに割り当てられた。各ゾーンのサイズは、メモリ・サイズの一部分であった。ホスト（ハイパバイザ）は、OSを監視したが、あらゆるゲストのゾーン内の任意の位置にアドレスできなければならなかった。

【0004】スモール・アドレスは、米国特許第4843541号に開示された従来の論理的に区分されたシステムで使用されている。このシステムでは、各ゲスト（guest）が異なるメモリ・ゾーンに配置され、かつ各ゲストが、IBM MVS（多重仮想記憶域）オペレーティング・システムやVM（仮想計算機）オペレーティング・システムのコピーなどのオペレーティング・システム（OS）となっている。ホスト（host）は、システムの異なる区画にあるすべてのゲスト・オペレーティング・システムを監視するシステム、たとえば（VMのような）ハイパバイザ・プログラムである。ゾーン（zone）は、システム主記憶装置内の別個の連続した区域であり、システムのすべての資源から構成された異なる論理区画に割り当てられる。このようなシステムを中央電子複合体（CEC）と呼ぶ。ゲストおよびホストはそれぞれ、31ビットのスモール・アドレス・サイズを使用する。論理的に区分されたシステムを制御するためのすべての既知のハイパバイザは、スモール・アドレスしか処理できない。そのようなシステムで使用されている「ウィンドウ」は知られていない。

【0005】「ウィンドウ（window）」は従来、パーソナル・コンピュータ（PC）で使用されており、PC DOSソフトウェア・オペレーティング・システムが使用する最下位1MBの記憶域における4つの未使用16KB区域に配置される。ウィンドウを使用して、最下位1MBのメモリと、1MBを上回る最大拡張アドレス$2^{24}-1$までの拡張メモリ・アドレスの間で、16KBページ分のデータが転送されていた。そのようなウィンドウによって、PCは最大16MBのメモリを使用することができた。そのような従来技術の例には、米国特許第4943910号がある。

6

【0006】大型メモリ・アドレスは、"Guest/Host Large Addressing Method and Means"と題する米国特許出願第8169111号（特願平4－329167号）に開示されている。この出願は、ゲストのスモール・アドレスまたはラージ・アドレスをラージ絶対アドレスに変換する方法を提供している。これは、アドレス指定にウィンドウを使用していない。

【0007】ラージ・アドレスを扱った別の特許出願には、"Large Logical Addressing Method and Means"と題する米国特許出願第803320号（特願平4－288025号）がある。この出願は、ゲスト・スモール・アドレスを使用して、大型メモリをアドレスするためのラージ・アドレスを生成している。これは、アドレス指定にウィンドウを使用していない。

【0008】

【発明が解決しようとする課題】メインフレームおよびその他のコンピュータのシステム・メモリのサイズを大幅に拡張すると、それらのコンピュータは、従来よりはるかに多くのデータおよびプログラムを格納し、従来使用していたよりも多くの機能を実行することができる。たとえば、メインフレーム内により多くの論理的区画を設けて、大型メモリにおけるゾーンの数を増やし、サポートできるゲスト・オペレーティング・システムの数を増やすことができる。

【0009】ゲストはホスト（ハイパバイザ）によって監視される。ホストは、（VMなどの）ソフトウェア、または（IBM PR／SM（プロセッサ資源／システム管理機能）などの）マイクロコードもしくは内部コードに存在することができる。ホストは、各ゲスト・ゾーン内の記憶位置にアクセスすることができなければならない。

【0010】現在のIBMメインフレームは、複数のゲストと1つのホスト（ハイパバイザ）との間で区分される小型のメイン・メモリ（最大2^{31}バイト＝2GB）を有する。現行のIBM PR／SMシステムおよびVMシステムでは、ホストは、必要に応じて、31ビットのスモール・アドレスを使用して任意のゲスト位置にアクセスする。このホスト要件により、メモリ・サイズが2^{31}バイト以下に制限される。なぜなら、ホストは任意のゲスト位置にアクセスできなければならないからである。ゲストも31ビット・アドレスを使用する。

【0011】現在使用している（スモール・サイズ・アドレスを使用するように設計された）ホストおよびゲス

トを変更してスモール・サイズ・アドレスを超えるラージ・サイズ・アドレスを使用できるようにするのは、コストがかかり面倒である。

【００１２】本発明の主目的は、システムのハードウェア／マイクロコードの変更を最小限に抑えてホストと大型メモリを使用する任意のゲストとの間でデータ転送を実行するのに必要なラージ・アドレスを従来型のホストが認識することなく、（スモール・サイズ・アドレスしか使用できない）当該ホストを大型メモリにおいてゲストと共に動作できるようにすることである。

【００１３】ゲストの識別信号をゲストのスモール・サイズ・アドレスと組み合わせてホストに与えることにより、ゲストの記憶位置をホストに示すことができる。

【００１４】したがって、本発明の別の目的は、ホストが従来型のスモール・アドレス（たとえば、３１ビット）を使用して、（たとえ、ゲストの記憶位置が、ホストが使用中のスモール・アドレスの範囲外である場合でも）大型メモリ内の任意のゲストの記憶位置にアクセスできるようにすることである。

【００１５】

【課題を解決するための手段】本発明によれば、ホストが、スモール・アドレスを使用して大型メモリ内の任意のゲストの記憶位置（ゲスト位置）にアクセスする能力は、次のようにして獲得される。１．ホスト・ゾーン内にウィンドウ・ページ・フレーム（ＰＦ）を定義する。２．ホストＰＦを、ホストに対して透過的なラージ・ゲスト・アドレスと関連付ける。ホストは、ゲスト位置をアドレスしたい場合、スモール・アドレスを使用してウィンドウＰＦにアドレスする。そうすると、ＣＰＵのハードウェア／マイクロコード（または内部コード）が、そのホスト・スモール・アドレスを、関連するラージ・アドレスで置き換え、その後は、ラージ・アドレスが、任意のゲスト・ゾーンにおける必要なゲスト位置を指定する。

【００１６】任意のゲスト位置に対する実ゲスト・アドレス（ラージ・アドレス）は、従来型のスモール・アドレスしか処理しないホスト・プログラムには見えない。（ホストおよびゲストに対して透過的な）ＣＰＵハードウェア／マイクロコードは自動的に、任意のゲスト位置がいつホスト・アテンションを受け取るかを検出し、ゲスト・ゾーンおよびそのゲスト位置に対するゲスト・スモール・アドレスをホストに対して識別する。この情報により、そのゲスト・ゾーンおよびスモール・アドレスを透過的に使用して、ホスト・プログラムが（ホスト・スモール・アドレスによって位置指定される）ホスト・ウィンドウを介して間接的に使用できるラージ絶対アドレスをセットアップするハードウェア／マイクロコードをホストが呼び出すことができる。

【００１７】したがって、ホスト・プログラムおよびゲスト・プログラムはそれぞれ、従来型のスモール・アドレスを使用しているものと仮定する。ＣＰＵハードウェア／マイクロコードは、スモール・アドレスの値に応じて、ゲストのゾーン原点にスモール・アドレスを加え、あるいはスモール・アドレスにプレフィクシング（prefixing）し、あるいはスモール・アドレスに逆プレフィクシング（reverse prefixing）することによって、ゲスト・スモール・アドレスをラージ・アドレスに変更する。

【００１８】従来型のホスト・プログラムがシステム上で本発明を使用するには、ホスト・プログラムを、ゲスト位置ではなくホスト・ウィンドウＰＦをアドレスできるように変更するだけでよい。ＣＰＵハードウェアおよびマイクロコードは、ホスト命令がスモール・アドレスを使用してゲスト位置をアドレスするとき、ホストがウィンドウＰＦを定義し、隠れたウィンドウ・レジスタ（ＷＲ）のゲスト・ラージ・アドレスを提供できるようにするだけでよい。

【００１９】本発明は次のように実施される。１．１つまたは複数のホスト・ウィンドウＰＦを位置指定するホスト・スモール・アドレスを含むウィンドウ・アクセス・レジスタ（ＷＡＲ）によって、ホスト・ゾーン内で各ホスト・ウィンドウＰＦを位置指定即ち特定する。２．各ホスト・ウィンドウＰＦを、ラージ絶対アドレスを収容するウィンドウ・レジスタ（ＷＲ）と関連付ける。

【００２０】ＷＡＲは、次の２つの代替方法のどちらかで実施される。１．１つまたは複数のＷＡＲを提供し、各ＷＡＲが単一のホストＰＦを位置指定する。２．単一のＷＡＲを提供し、このＷＡＲの内容が複数の連続するホストＰＦを位置指定する。これらの複数のＰＦの中から、（複数の連続するホスト・ウィンドウＰＦの組における）特定のホスト・ウィンドウＰＦが、ホスト・ウィンドウＰＦにアクセスするために提供された各ホスト・スモール・アドレスにおけるＰＦアドレスの下位ビットによって選択される。複数の異なるホスト・ウィンドウＰＦのアドレスは、単一のＷＡＲで定義されるときでも、ホストが独立に使用することができる。すなわち、ホストは、同一のゲスト・ゾーンまたは複数のゲスト・ゾーンにおける異なるゲスト位置にアクセスするために、複数のウィンドウＰＦを同時に使用することができる。

【００２１】各ＷＡＲの内容は、ホストによって指定されるホスト・ゾーンのウィンドウＰＦを位置指定するスモール・アドレスをＷＡＲにロードする命令を実行するホストによって構成される。

【００２２】１つまたは複数のＷＲの存在、並びにＷＲとホスト・ウィンドウ（ＷＲと同数）との関連付けは、システムのハードウェア／マイクロコードの構造によって決まる。ただし、ＷＲとホスト・ゾーンＰＦとの関連付けは、「ＷＡＲロード（LOAD WAR）」命令が実行されるまで決定されず、ＷＲと特定のゲスト・スモール・ア

ドレスとの関連付けは、「ウィンドウ活動化（Activate Window）」命令が実行されるまで決定されない。

【0023】さらに一般的に言うと、単一のWARによってゾーン内で位置指定される連続するウィンドウPFの数は、2^P（Pは、ウィンドウにアクセスするためのゾーン・オペランド・アドレスにおけるPFコンポーネントの最下位ビットの数を表す）によって定義される。ここで、したがって、ゾーン・オペランド・アドレスは、命令によってホスト・ゾーン内に提供され、そのようなアドレスは、指定されたホスト・ウィンドウPFセットにおける必要なウィンドウPFを自動的に選択する。これは、ゾーン・オペランド・アドレスが、ウィンドウPFの選択を自動的に実行するP個の最下位ビットを持つ「PFアドレス・コンポーネント」を含むからである。

【0024】ゾーン内の各ウィンドウは、ウィンドウ・レジスタ（WR）およびウィンドウ空間と関連付けられる。ウィンドウ空間とウィンドウの関連付けは、ホスト・ゾーンで実行されるWARロード命令によって明示的に実行される。WRとウィンドウの関連付けは、本発明を提供するCPUハードウェアまたはマイクロコードの設計中で暗示的に実行される。

【0025】実行するホスト命令オペランドが任意のホスト・ウィンドウPFアドレスに等しいとき、CPU内のハードウェア／マイクロコードが、そのホスト・ウィンドウPFに関連付けられたWRに含まれるラージ絶対アドレスを自動的に置き換える。この動作によって、関連付けられたWR中にラージ・アドレスを持つゲスト位置に対する、ホスト・オペランドによる自動的な間接ラージ・アドレス指定が実行される。

【0026】ホスト命令オペランド・アドレスがどのホスト・ウィンドウPFアドレスとも等しくないときは、ホスト・ウィンドウは選択されず、間接ラージ・アドレス指定動作は行われない。この場合、ホスト命令オペランド・アドレスは、ホスト・ゾーンにおいて、従来技術の場合と同様に、プレフィクシングおよび逆プレフィクシングを含めて、通常のホスト・アドレスとしてアクセスされる。

【0027】記憶域オペランドを持つどんな種類のホスト命令も、ホスト・ウィンドウ・アドレスを指定して、関連するゲスト・アドレスにアクセスすることができる。どんな種類のホスト命令動作もゲスト・ゾーンで行うことができる。たとえば、ホストは、それぞれ2つの異なるWRが関連付けられている2つの異なるゲストの2つのオペランド用に2つの異なるホスト・ウィンドウPFを指定することによって、2つの異なるゲスト位置の間で記憶域オペランドを移動することができる。

【0028】したがって、本発明の他の目的は、ホストがスモール・アドレスを使用して、2つの異なるゲスト位置の間でデータを移動／コピーし、あるいは同一のゲスト・ゾーン内の2つの記憶位置の間でデータを移動／コピーできるようにすることである。

【0029】ゾーン情報テーブル（ZIT）は、システムに常駐するそれぞれのゲストのゾーンを定義する。ハードウェア／マイクロコードでサポートされるWR活動化プロセスは、要求側ゲストのスモール・アドレスを、それぞれのゲストのゾーン用のZITエントリ中のゾーン限界情報と突き合わせて検査して、スモール・アドレスがゲストのゾーンのアドレス指定範囲内にあることを確認し、その後に、ラージ・アドレスをWRに書き込む。

【0030】WRに（ゲストのスモール・アドレスを表す）ラージ・アドレスをロードするためのプロセスは、システム内の仮想マシンのインタセプト事象または割込み事象の検出によって呼び出すことができる。そのような事象の存在は、SIE（解釈実行開始）命令の状態記述（SD）に記憶された信号によって示すことができる。システム内の各ゲストにはそれぞれSDが割り当てられる。SDとは、システムが各ゲストの動作を処理するのに必要とするすべての情報を記憶する主記憶装置内の制御ブロックである。ホストに利用可能なSD内のインタセプト、割込み、およびそれらのスモール・アドレスを示す方法は、"IBM System/370 Extended Architecture Interpretive Execution"と題する1985年のIBM刊行物SA22－7095－1に詳細に記載されている。

【0031】本発明で使用できる、任意のSD内の重要な情報としては、インタセプト事象および割込み事象の発生を示す標識、および現インタセプト事象または割込み事象の影響を受ける、ゲスト・ゾーン内の記憶位置を指定するスモール・アドレスがある。

【0032】ホストは、ゲストのインタセプト指示または割込み指示に応じてゲストのSDからホストに渡される情報に基づき、ゲスト・ゾーン内の、ホストによってアクセスされる各ゲスト・オペランドにホスト・ウィンドウを割り当てる。

【0033】本発明のもう1つの特徴は、システム動作中の任意の時にZIT（ゾーン情報テーブル）を再構成することによって、システム内のホストおよびゲスト用のゾーンを動的に再定義できることである。

【0034】ホストは、任意のゲストのゾーンにアクセスする固有の権限を持つ。この権限は、ホストが、ゲストの処理によって誤って発生する場合がある予測できない問題からゲストを保護するのに肝要である。

【0035】ホスト・ゾーンが、大型メモリ内の（最下位アドレス範囲を持つ）メモリの第1ゾーンである必要はない。ホスト・ゾーンは、大型メモリに適用されるどのスモール・アドレス指定機能の範囲外であってもよい（これは、PC拡張メモリをサポートするために使用される従来技術のPCウィンドウとは異なる）。

【００３６】本発明により、コンピュータ・システム内でハードウェアまたはマイクロコードを単純に追加することによって、スモール・アドレスを使用するホスト・プログラムが、大型メモリ内の任意の位置にアクセスすることが可能となる。

【００３７】

【実施例】

大型メモリのマップ－図１

本発明の好ましい実施例では、メインフレームの大型メモリ内に１つのホスト（ハイパバイザ）と複数のゲスト（オペレーティング・システム）が配置され、メモリ・サイズは２ギガバイト（２ＧＢ）を上回る実記憶域であり（２ＧＢの数倍であることが好ましい）、ホストは、２ＧＢの範囲内しかアドレスできない３１ビットのスモール・アドレス（small address）を使用する。

【００３８】図１は、大型メモリ内の領域を１つのホスト、複数のゲスト、および「システム・エリア記憶域」（すべて、システム主記憶域ハードウェア・エンティティの一部）に割り当てたマップを示す。しかし、システム・エリア記憶域にアドレス指定可能なのはハードウェアおよびマイクロコード（内部コード）だけであり、システム内のＣＰＵ上で走行するプログラムによってアドレスすることはできない。ホスト・ゾーンおよびゲスト・ゾーンはそれぞれ、メモリ・エンティティのプログラム・アドレス指定可能部分に配置されている。各ゾーンのサイズは、2^{31}バイト未満である。

【００３９】各ゾーン（ホストまたはゲスト）は、メモリ・エンティティ内で空間が存在する任意の場所に配置することができ、かつメモリ・エンティティ内に任意の順序で配置することができる。各ゾーン内のメモリ空間は連続している。

【００４０】ホストおよび各ゲストは、スモール・アドレスを使用する。好ましい実施例では、スモール・アドレスは３１ビットのサイズを有する。スモール・アドレスは、単一のゾーン内の記憶位置にしかアドレスできない。これは、各ゲストによって発行される各メモリ・アドレスに対する限界検査によって実施される。

【００４１】実メモリ空間は、「ページ・フレーム（ＰＦ）」と称する単位に分けられる。ページ・フレームは、アドレス変換および再配置に使用される。好ましい実施例では、実メモリ空間が４キロバイト（ＫＢ）のページ・フレーム（ＰＦ）に分けられ、メモリ内の各ＰＦが４ＫＢ境界上に配置される。

【００４２】従来の方法では、まずゲストの各仮想アドレスが実アドレスに変換される。次に、実アドレスが、ＩＢＭ　ＥＳＡ／３９０システム用の従来技術による方法でアドレス・プレフィクシングを使用することによって絶対アドレスに変換される。この場合、ゼロの実アドレスと、ＣＰＵのプレフィクス・アドレスに等しい実アドレスとを除き、すべての実アドレスは、それに対応する絶対アドレスと同じである。

【００４３】本明細書では、すべてのアドレスは、メモリ・アクセスに使用する際は絶対アドレスである。

【００４４】好ましい実施例では、ホスト・ゾーン内に２つのウィンドウを提供し、これらのウィンドウを使用して、ラージ・アドレス（large address）を必要とする大型実メモリ内における、他の任意の１つのゾーン内もしくは他の任意の２つのゾーン内の任意の２つの記憶位置にアクセスする。

【００４５】ホストは、そのゾーン内の任意の場所にこれらの２つのウィンドウを配置することができる。ウィンドウは、ホスト・ゾーン内のデータにアクセスするホスト命令オペランドに妨害を与える可能性が最も少ない場所に配置することが好ましい。

【００４６】各ウィンドウには、ホスト・ゾーン内のページ・フレーム（ＰＦ）が割り当てられる。好ましい実施例では、１つのウィンドウ・アクセス・レジスタ（ＷＡＲ）によって２つのウィンドウを位置指定する（locate）。この割当ては、図２に示す「ＷＡＲロード（load WAR）」命令を実行することによって行う。

【００４７】しかし、図３には「ＷＡＲロード」命令の別の実施例が示されている。この場合、複数のＷＡＲが、ホスト・ゾーンにおけるそれぞれのウィンドウとして、不連続のＰＦを位置指定する。この実施例では、図３に示す「ＷＡＲロード」命令を使用するが、この命令は複数のＷＡＲのうちの１つに単一のホストＰＦを割り当てる。したがって、複数のＷＡＲの各々にホストＰＦを割り当てるために、ＷＡＲの数と同じ数のＷＡＲロード命令を実行する必要がある。

【００４８】ＷＲ自体は、ホストまたは任意のゲストにとって明らかである必要はない。ＷＲの存在を、ホストまたはゲストのいずれかのプログラム・コードで示す必要はない。

【００４９】ここに示す好ましい実施例では、ホスト・ゾーンだけがウィンドウを使用し、ゲスト・ゾーンはウィンドウをもたないものと仮定する。ＷＡＲは、システムの初期設定時にロードされる。

【００５０】単一のＷＡＲが２つのウィンドウＰＦを位置指定する好ましい実施例では、ホスト・オペランドのＰＦアドレス・コンポーネントの最下位ビットの状態が、これらの２つのウィンドウＰＦのうちどちらをアドレスするかを指定する（４つのホスト・ウィンドウＰＦが提供されている場合は、オペランド・アドレスＰＦコンポーネントの最下位２ビット（３１ビット・アドレスのビット１８およびビット１９）の状態を使用して、アドレスされるウィンドウＰＦを選択する）。

【００５１】さらに詳細に述べると、ホストに２つのウィンドウ番号が提供され、各ウィンドウ番号は、暗示的に関連付けられたＷＲを持つ。ＷＲは、マシンの設計時に、またはマシン・マイクロコードのあるバージョンが

マシンにロードされるときに固定される。好ましい実施例で指定されるWR番号はウィンドウ番号に順次固定され、ウィンドウ1にはWR0が、ウィンドウ2にはWR1が割り当てられる。ウィンドウ番号がこれより多い場合、それらの番号はそれぞれ異なるWRと関連付けられる。したがって、各ウィンドウ番号は、暗示的に関連付けられたWRを持つ。

【0052】好ましい実施例ではまた、WRはそれぞれ、大型メモリ内の任意の位置にアクセスするために63ビット・アドレスを含むことができる。

【0053】ウィンドウ数が固定され、WRとホスト・ゾーンが利用できるウィンドウとの関連付けが固定されているので、ホストはWARロード命令を使用して各ウィンドウ番号にウィンドウ空間を割り当てる必要がある。

【0054】ホスト・ゾーンにおけるウィンドウの位置指定－図2

各ホスト・ウィンドウに対するホストPFの割当ては、ホストで実行される「WARロード」命令によって行う。このホストによるPFおよびウィンドウ間の割当てでは、PFとウィンドウを関連付けるプロセスで、WRが、指定されたPFに暗示的に割り当てられる。WARロード命令の実行は、ホストが実行中のシステム内の任意のCPUのハードウェア／マイクロコードによって行われる。

【0055】図2は、WARロード命令用のハードウェアおよびプロセスを示す。この命令は、命令コードを持ち、汎用レジスタR1を指定する単一のオペランドを持つ。命令実行前に、R1のビット位置1ないし18にホストPFアドレスがロードされている。このアドレスは、2つの連続するPFを含む8KBのエリアを位置指定する。割当てプロセスにより、R1のビット1ないし19のPFアドレスが、WARのビット位置1ないし19に移され、WARのビット位置20ないし31にはゼロが書き込まれる。WR0およびWR1は2つのウィンドウと暗示的に関連付けられる。WARでアドレスされるPFはWR0と関連付けられ、もう一方のPFはWR1と関連付けられる。オペランド・アドレスのビット19の状態によって、適切なウィンドウPFが選択される。

【0056】ホストは、各CPU上で、システム内の他のCPUとは独立して「WARロード」命令を実行するので、各CPUにはそのWARがロードされる。希望するなら、異なるCPU内で、複数のホストPFを同一のウィンドウ番号には、割り当てることができる。本明細書での議論では、異なる割当てによって生じる混乱を最小限に抑えるために、すべてのCPUにおいて同じウィンドウ番号には、同一のホストPFアドレスが割り当てられるものと仮定する。

【0057】ホストがゲスト位置にアクセスできるようにするには、WRにゲスト・ラージ・アドレスをロードしておく必要がある。任意のゲスト位置にアクセスするために、ホストはどちらかのホスト・ウィンドウを使用することができる。WRにゲスト・ラージ・アドレスをロードした後、ホスト・ソフトウェアは、WARによって現在指定されている任意のウィンドウPF内のバイト位置に対するオペランド・アドレスを持つ任意の命令を実行することができる。その後、ハードウェアは、そのホスト命令を実行する際にホストのスモール・アドレスをラージ・アドレスで置き換えることによって、関連するWR内のラージ・アドレスに暗示的にアクセスする。

【0058】アクセスされるゲスト位置は、どのゲスト・ゾーンのどのバイト境界上にあってもよい。各ホスト・ウィンドウは、PF境界上に配置される。これは、任意のホスト・アドレス用のすべての可能なバイト変位（D）値をウィンドウに収容するために、メモリ空間のPFが必要だからである。

【0059】ウィンドウに対する各ホスト・アドレスはアドレス置換動作を引き起こすので、ホストは、「ウィンドウ活動化」命令でウィンドウが活動化されている間に、任意のホスト・ウィンドウPF宛のどのアドレスを使用しても、任意のホスト・ウィンドウに記憶されたデータを取り出し、あるいはそこに記憶することはできない。したがって、ホストは、ウィンドウのアドレスを使用してアクセスされると予想されるそのどのウィンドウPFにもデータを記憶すべきではない。

【0060】しかし、ホストは、そのデータを収容するための1つまたは複数のウィンドウPFを含む、複数の連続するPFを必要とするラージ・オペランド用に非ウィンドウ・アドレスを使用する場合、任意のウィンドウPFにデータを記憶し、そこから取り出すことができる。たとえば、move-character-long命令中のオペランドは、1つまたは複数のウィンドウPFに及ぶオペランド・データ用の非ウィンドウ・オペランド・アドレスを指定することができる。

【0061】ホストによるゲスト・アクセス用のウィンドウ割当て－図3

ホストは、「ウィンドウ活動化」命令を実行することによって、ゲスト位置にウィンドウ番号を割り当てる。この命令は、ゲスト・ゾーン（R3レジスタ中でホストによって指定される）内のゲスト・スモール・アドレス（R1レジスタ中でホストによって指定される）にウィンドウ番号（R3レジスタのフィールド中でホストによって指定される）を割り当てる。次に、指定されたウィンドウ番号と関連するWRが、その指定されたゲスト位置と暗示的に関連付けられる。この命令は、対応するゲスト・ラージ絶対アドレスを暗示的に算出し、そのアドレスを、関連するWRと、「ウィンドウ活動化」命令を実行するCPUのCPUウィンドウ指定テーブル内のそのWR用のエントリとに入力する。

【００６２】ホストは、ＷＲの存在を知る必要がなく、また、ウィンドウを使用する際にＷＲのラージ・アドレス内容にアクセスできる必要もない。ホストが知る必要があるのは、そのウィンドウ番号だけである。ホスト・ソフトウェアに対して透過的な自動ハードウェア／マイクロコード動作によって、ラージ・アドレスが生成され、ホストが「ウィンドウ活動化」命令を実行するとき、ＷＲにこのラージ・アドレスが入れられ、ホストがホスト・ウィンドウを参照する命令を実行するとき、自動的に、ホスト・オペランドにおけるスモール・ウィンドウ・アドレスがラージ・アドレスで置き換えられる。

【００６３】ハードウェア／マイクロコードを使用してゲストのラージ・アドレスを生成しＷＲに入れるために、ホストが「ウィンドウ活動化」命令を呼び出すことができるようにするには、ホストによる準備動作が必要である。この準備動作には、ＳＤテーブル（図９に表す）にゲストのインタセプト事象および割込み事象があるか否かをホストに監視させることが含まれる。ホストによる監視は、ＣＰＵの信号によって条件付けることができる。この信号は、インタセプト事象または割込み事象が発生したとき、または、ゲストのいずれかのＳＤエントリ中におけるインタセプト指示または割込み指示を検出するためにＳＤテーブルを監視するタイマ割込みによって、ホストがそれ自体に定期的に割り込んだときに、レジスタＲ１およびＲ３をロードするためにホストの監視動作を呼び出す。

【００６４】ホストは、テーブルのゲストＳＤ中でそのような指示を見つけると、図３に示す「ウィンドウ活動化」命令プロセスを実行する前に、そのＳＤから、Ｒ１汎用レジスタ４２およびＲ３汎用レジスタ４３の内容をセットアップするための情報を読み取ることができる。

【００６５】Ｒ１をセットアップするには、ホストがゲスト・アクセスにそのどちらのウィンドウを使用するかを決定し、そのウィンドウ番号をＲ１に入れる。ゲスト・ゾーン番号（ＺＮ）は、インタセプトされたＳＤエントリに対するインデックスとして、あるいはそのエントリ内のＺＮフィールドを読み取ることによって、当該ＳＤエントリによって取り出される。このＺＮ値は、ホストによってＲ１汎用レジスタ４２のＺＮフィールドにロードされる。

【００６６】Ｒ３をセットアップするには、アクセスされるゲストのアドレスが、（インタセプト事象または割込み事象の発生時にマイクロコードによってＳＤに入れられた）インタセプトされたＳＤの別のフィールドから取り出され、そのアドレスがホストによってＲ３汎用レジスタ４３にコピーされる。

【００６７】その後、図３に表したプロセスを使用してウィンドウ活動化命令を実行することができる。この命令は、Ｒ１にある３１ビットのゲスト・スモール・アドレスから６３ビットのゲスト・ラージ・アドレスを生成し、その６３ビット・アドレスを、ホストで選択されたＷＲにロードし、またその６３ビット・アドレスを、ＣＰＵウィンドウ指定テーブル・エントリ４７内でそのＷＲに割り当てられたエントリの「ラージ絶対アドレス」フィールドにロードする。各ＣＰＵに「ＣＰＵウィンドウ指定テーブル」があり、このテーブルは、プログラムがアドレス指定可能でない「システム・エリア記憶域（図１）」に存在することができる。このＣＰＵテーブルは、それぞれのＣＰＵ内の各ＷＲの内容を表すエントリを持つ。ＷＲにロードされる６３ビット・アドレスは、ゲスト・ゾーン内のバイト・アドレスである。図８は、関連するＣＰＵ内の各ＷＲ用のそれぞれのＷＲエントリを含む、各ＣＰＵウィンドウ指定テーブルを表している。各テーブル・エントリは、ゲストのゾーン番号と、ゲストのスモール・アドレスと、ＣＰＵテーブル・エントリ中のスモール・ゲスト・アドレスを表すゲストのラージ絶対アドレスとを含む、ＷＲへの現ゲスト・アドレス割当てを含んでいる。

【００６８】Ｒ１中のＺＮ値は、アクセスされるゲストのゾーンと関連するＺＩＴエントリにアクセスするためにＺＩＴ　４４に索引付けするのに使用される。加算機構４９は、６４ビット加算機構であり、Ｒ３レジスタ４３中のスモール・ゲスト・アドレスに、ＺＩＴエントリから得られたゾーン原点を加算する。この結果、６３ビットの絶対アドレスが生成され、このアドレスが比較機構４５に提供される。比較機構４５は、生成された絶対アドレスを、ＺＩＴエントリから得られたゾーン限界と比較する。絶対アドレスがＺＩＴエントリのゾーン限界より小さい場合（通常の場合）、Ｒ１汎用レジスタ４２が供給するＷＲ番号で位置指定されたＣＰＵウィンドウ指定テーブル・エントリ４７にこの絶対アドレスが格納される。絶対アドレスがゾーン限界より大きい場合、レジスタ４６によってＣＰＵウィンドウ指定テーブル・エントリ４７に例外スモール・アドレス値が出力され、例外信号が生成されてＣＰＵに割り込む。ＣＰＵウィンドウ指定テーブル・エントリ４７にロードされた絶対アドレス値は、次にウィンドウ・レジスタ４８にロードされる。ウィンドウ・レジスタ４８はこのときラージ・アドレスを含んでおり、関連するウィンドウＰＦがホスト命令によってアドレスされるとき、それがゲスト・スモール・アドレスと置き換わる。

【００６９】ＣＰＵウィンドウ指定テーブル・エントリ４７は、図７に関連して説明する、ＺＩＴ内でのゾーン再配置中に使用される。ゾーン再配置は、特殊コマンドによって開始される。

【００７０】ＺＩＴを図１に示す。各ゾーンはＺＩＴエントリを有し、ＺＩＴエントリは、ゾーン番号、ゾーン原点、およびゾーン限界を含むフィールドを有する。エントリ内のゾーン番号を使用して、ＺＩＴのアクセスに使用されるＲ１のＺＮ値を検証することができる。

【００７１】ＷＡＲレジスタおよびＷＲレジスタがＣＰＵの動作に影響を及ぼすのは、システム（ＣＥＣ）が区画モードで動作しており、ＷＡＲが非ゼロ値を含むときだけである。

【００７２】ゲスト位置にアクセスするためのプロセス－図４および図５

ホストは、割込みまたはインタセプトの処理中に検出されたゲスト位置へのアドレス指定可能性を獲得するために、図４または図５のハードウェアおよびプロセスを使用する。ゲスト位置は、ホストがゲスト命令をシミュレートするためにしばしばアクセスする。図４では１つのＷＡＲを、図５では複数のＷＡＲを使用しているが、その他の点では、どちらの場合も同様に動作する。

【００７３】インタセプト・コードは、インタセプトされたゲスト命令をホストが検査しなければならないことを示すことができる。ホスト・ソフトウェアは、ゲストのＳＤエントリにアクセスし、ゲストのゾーン番号（ＺＮ）と、インタセプトされた命令のスモール・アドレスとを、ＰＳＷ（プログラム状況ワード）フィールドに取り込む。ホストは次に、ウィンドウ番号を指定し、ウィンドウ活動化命令を実行して、ゲスト位置へのアドレス指定可能性を獲得する。ホストは次に、ゲスト・ゾーン内のゲスト命令にアクセスし、ホスト・ゾーン内のホストにより指定される位置にこの命令をコピーするためのホスト命令に、各ゲスト・オペランド用の関連するゲスト・ウィンドウＰＦを入れる。

【００７４】さらに詳細に述べると、ホストは、このアクセスを実行するために、そのウィンドウ番号のうちの１つを選択し、選択したウィンドウ番号をＲ１汎用レジスタ４２に入れ、ゲストのＺＮとＰＳＷアドレスをＲ３レジスタ４３に入れ、次いでウィンドウ活動化命令を呼び出して、ラージ・アドレスを生成し、選択したウィンドウ番号と関連するＷＲにそれを格納する。このようにして、ホストは、図６の流れ図に示すプロセスを実行する際に図４または図５に表されたハードウェアを使用して、ゲスト命令にアクセスする。

【００７５】図４または図５で、スモール（実）アドレスが、ゲストのＳＤへのホスト・アクセスによって、入力レジスタ５４１または６２１に入力され、対応するラージ絶対アドレスが出力レジスタ５５１または６２９に出力される。

【００７６】ホスト・ウィンドウがホスト命令によってアクセスされるとき、図４または図５で使用されるアドレス選択プロセスは、複数の種類のラージ・アドレスのうち１種類を出力レジスタ５５１または６２９に出力する。ラージ・アドレスの種類には、ゲスト・ゾーン・プレフィクス絶対アドレス、ゲスト・ゾーン逆プレフィクス絶対アドレス、またはプレフィクス・アドレスでも逆プレフィクス・アドレスでもないゲスト・ゾーン絶対アドレスがある。

【００７７】ホストが、そのどのウィンドウに対するものでもない記憶域オペランドを指定するとき、そのオペランドはホスト・ウィンドウ内でアクセスされる。すなわち、ホスト命令が、ＷＡＲによってホスト・ウィンドウと定義されていないホストＰＦにアクセスするとき、図４または図５で使用されるアドレス選択プロセスは、出力レジスタ５５１または６２９にホスト・ゾーン・ラージ絶対アドレスを出力する。

【００７８】入力されたスモール・アドレスが実アドレス・ゼロである場合、図４または図５においてプレフィクシング・プロセスが動作して、それぞれのＣＰＵのプレフィクス・レジスタ中に提供された所定の非ゼロ絶対アドレスを実アドレス０で置き換える。システムの各ＣＰＵは、そのプレフィクス・レジスタ（図４または図５のレジスタ５４４）中に事前に割り当てられた異なるプレフィクス・ラージ・アドレスを持っているので、（割込み値など、そのＣＰＵだけに関する値を含む）異なるプレフィクス・ページ・フレームにアクセスする。プレフィクス変換動作で、そのプレフィクス・レジスタ中のプレフィクス・アドレスをスモール実アドレス０で置き換えることによって、絶対アドレスが生成される。また、図４または図５で、ラージ・アドレス・ゼロをＣＰＵのプレフィクス値に等しい任意のスモール実アドレスで置き換える逆プレフィクシング・プロセスが実行されるので、逆プレフィクス動作では必ずすべてのＣＰＵについてページ・フレーム０にアクセスする。

【００７９】図４および図５のそれぞれで、入力レジスタ５４１または６２１中に入力されたホスト・スモール実アドレスのビット１ないし１９が、（プレフィクシングのために）比較機構５４２でゼロと比較され、同時に（逆プレフィクシングのために）比較機構５４３で、ＣＰＵのプレフィクス・レジスタ５４４中のビット１ないし１９によって提供されるプレフィクス値と比較される。比較機構５４３または５４２がcompare equal条件のとき、プレフィクス・レジスタ５４４中のラージ・アドレスまたは逆プレフィクス・ゼロ・ラージ・アドレスは、それぞれ選択機構５５０または６２８に送られ、選択機構５５０または６２８は、それぞれのラージ・アドレスを出力レジスタ５５１または６２９に出力する。

【００８０】図４と図５は、ホスト・ウィンドウＰＦが選択される方法が異なる。図４と図５のどちらでも、ホスト・オペランド・スモール・アドレスの値が、ＷＲの選択を制御する。図４では、ホスト・ウィンドウＰＦの選択が、そのホスト入力レジスタ５４１中のビット１９の状態によって決定される。すなわち、入力レジスタのビット１９の状態が０のときは、ＷＲ０のウィンドウが選択され、ビット１９の状態が１のときは、ＷＲ１のウィンドウが選択される。図５では、各ＷＲがそれぞれ異なるＷＡＲと関連付けられ、（それに含まれるホスト・ウィンドウＰＦのビット１ないし１９が入力ビット１な

いし１９に等しいＷＡＲと関連付けられた）ＷＲが選択される。

【００８１】図４でさらに詳細に説明すると、スモール・ホスト実アドレスをラージ絶対アドレスに変換する手順は、次のプロセスによって示される。

【００８２】１．入力アドレス・ビット１ないし１８がＷＡＲ　５４５中のビット１ないし１８に等しく、入力レジスタのビット１９が０であり、ＷＲ０の内容がゼロに等しくないときは、ＷＲ０で指定される絶対ページ・フレーム・アドレスで、入力アドレスのビット１ないし１９が置き換えられる（ラージ絶対アドレス・ビット３３ないし５１を置き換える）。ＷＲ０中の最上位ビット１ないし３２は影響を受けない。これらのビットは、図３で提供されるゾーン原点ビットのままである。ＷＲ０中の最下位ビット５２ないし６３も影響を受けない。これらのビットは、入力されたスモール・アドレスからの変位ビット２０ないし３１である。

【００８３】２．入力アドレス・ビット１ないし１８がＷＡＲビット１ないし１８に等しく、入力レジスタ・ビット１９が１であり、ＷＲ１の内容がゼロに等しくないときは、ＷＲ１で指定される絶対ページ・フレーム・アドレスがアドレスのビット１ないし１９と置き換わる。

【００８４】３．入力された実アドレス・ビット１ないし１８がＷＡＲビット１ないし１８に等しくないとき、あるいは入力されたビット１９が０であり、ＷＲ０の内容がゼロに等しい場合、あるいは入力されたビット１９が１であり、ＷＲ１の内容がゼロに等しい場合、アドレスは元のままである。

【００８５】ＷＡＲの内容がゼロに等しいとき、出力されるラージ・アドレスにおいて、入力されたアドレスは元のままである。なぜなら、ホスト・ウィンドウが選択されていないからである（ホスト・ウィンドウ０はウィンドウとして使用できない）。この場合、出力されるアドレスは、入力されたアドレスにおいて、ホスト・ゾーン原点がスモール・アドレスの最上位端と連結されたものである。

【００８６】いずれの場合も、入力レジスタ５４１のスモール・アドレスのＤビット２０ないし３１は、変更されずに、出力されるラージ・アドレスに移される。

【００８７】ＷＡＲのビット１ないし１９がホスト・プレフィクス・レジスタのビット１ないし１９に等しいときは、プレフィクス・レジスタ５４４中のスモール絶対アドレスがラージ絶対アドレスとして出力される。

【００８８】図４では、選択機構５４８がスモール実アドレス５４１の最下位ビット１９を使用して、２つのウィンドウ・レジスタ５４６および５４７の一方を選択する。ＷＡＲによって３つ以上のホスト・ウィンドウＰＦが提供される場合、選択機構５４８は、入力されたスモール・アドレスのビット１９を通じて複数の最下位ビットを受け取り、それぞれがウィンドウＰＦと関連するｎ個のＷＲのうちの１つを選択する。

【００８９】図５では、選択機構６２７は図４の選択機構５４８と同様に動作する。しかし、図５では、選択機構６２７は入力されたスモール・アドレス・ビット１ないし１９を各ＷＡＲのビット１ないし１９と比較し、等しいビット１ないし１９を持つＷＡＲのＷＲを選択する。

【００９０】図４では、別の選択機構５５０が、実アドレス５４１、プレフィクス・レジスタ５４４、選択機構５４８で選択されたウィンドウ・レジスタ、およびゼロの値のうちの１つを次のように選択する。入力レジスタ５４１中の入力された実アドレスがゼロに等しい（とボックス５４２で判定された）場合、プレフィクス・レジスタ５４４が選択される。実アドレス５４１がプレフィクス・レジスタ５４４に等しい（とボックス５４３で判定された）場合、ゼロの値が選択される。入力レジスタ５４１中の入力されたビット１ないし１９がＷＡＲ　４５のビット１ないし１９に等しい（とボックス５４９で判定された）場合、選択機構５４８で選択されたＷＲ値が出力される。これらの比較がいずれも等しくならない場合、レジスタ５４１中の入力されたアドレス・ビット１ないし１９が、ホスト・ゾーンにアクセスするため、ラージ・アドレス・ビット３３ないし５１として出力選択機構に直接入れられる。

【００９１】図５では、選択機構６２８は図４の選択機構５５０と基本的に同じ動作をする。

【００９２】選択機構５５０または６２８中のラージ絶対アドレスの値は最終的に、出力レジスタ５５１または６２９に移され、このレジスタから、アドレスを使用して大型メモリがアクセスされる。

【００９３】ホストによるゲスト位置へのアクセス―図６

図６は、図４または図５に示すハードウェアがラージ絶対アドレスを生成するのに使用するプロセスを表している。各図で、入力されたスモール実アドレスがラージ絶対アドレスに変換される。アドレス変換プロセス中、処理のプレフィクス／ゼロ部分はＷＲ／ＷＡＲ選択処理と並行して実行することができる。

【００９４】処理は、ステップ７７１から開始する。ステップ７７２に進み、入力アドレスのビット１ないし１９がゼロの値と比較される。入力ビット１ないし１９がゼロに等しい場合、ステップ７７３で、絶対アドレスがプレフィクス・レジスタの内容に設定される。ステップ７７２で入力ビット１ないし１９がゼロに等しくないと判定された場合、ステップ７７４で、これらの入力ビットが、プレフィクス・レジスタの対応するビットと比較される。ステップ７７２でこれらの入力ビットどうしが等しい場合、ステップ７７５で、絶対アドレスがすべてゼロに設定される。ステップ７７４でこれらのビットが等しくないと判定された場合、ステップ７７６で、入力

ビット１ないし１８がＷＡＲビット１ないし１８と比較されて、ウィンドウが使用中かどうかが判定される。

【００９５】ビット１ないし１８が等しい場合、ステップ７７８で、入力されたアドレスの下位ビット１９が復号されて、ＷＲ０とＷＲ１のどちらを選択するかが決定される。

【００９６】次に、ステップ７７９で、入力アドレス・ビット１ないし１９が、絶対アドレス中の適切なウィンドウ・レジスタ・ビット１ないし５１と置き換えられて、出力アドレス７８１が提供される。

【００９７】ステップ７７６で入力されたビット１ないし１８が、ＷＡＲ中の対応するＰＦビットと等しくないと判定された場合、ステップ７７７に進み、入力されたアドレス・ビット１ないし１９が絶対アドレス・ビット位置３３ないし５１に格納され、これによって出力される絶対アドレス７８０が提供される。

【００９８】絶対アドレス７８０または出力アドレス７８１は、要求されたアクセスを行うために、大型メモリの制御装置に提供される。

【００９９】ゾーンの再配置－図７

図７は、１つまたは複数のＣＰＵを持つことが可能なシステム内で大型メモリを再構成するためのＣＰＵゾーン・アドレス再配置のフロー・チャートを表す。ゾーンは、システム内のホスト区画および論理区画の間で再構成される。大型記憶域におけるゲスト・ゾーンまたはホスト・ゾーンの位置の変更は、ホスト・プログラムおよびゲストに対して透過的である。任意の既存のゲスト・ゾーンの原点または限界が変更され、任意のゲスト・ゾーンが削除され、１つまたは複数の新規ゲスト・ゾーンが追加される可能性がある。ゲスト・ゾーン原点が変更される場合、そのゾーンが大型メモリ中で再配置される。この場合、ＷＲ０またはＷＲ１中のラージ絶対アドレスのどちらか一方または両方を変更しなければならない。

【０１００】ホスト・ゾーンは暗示的（ＺＩＴ中で定義されない）でも、明示的（ＺＩＴ中で定義される）でもよい。暗示的な場合、ホスト・ゾーンが再配置されることはなく、ホスト限界の検査は行われない。明示的な場合、各ＷＡＲの内容と、ＺＩＴホスト・エントリ（たとえば、ＺＮ０をもつエントリ）中のホスト・ゾーン限界の間で検査が行われる。

【０１０１】任意のゲスト・ゾーン限界が変更される可能性があるので、（ホスト・ウィンドウが使用中の）各ＣＰＵウィンドウ指定テーブル・エントリ中の各スモール・アドレスを、ゲスト・ゾーン限界と突き合わせて検査する必要がある。ここで、限界を超えている場合、（ＣＰＵウィンドウ指定テーブル・エントリ中、および対応するＷＲ中の）スモール・アドレスと、それに対応するラージ絶対アドレスを無効にしなければならない。

【０１０２】再構成を実行するには、システム内のＣＰＵのうちの１つによって特殊コマンドが発行され、ステップ９１でゾーン再配置プロセスが開始する。このコマンドは、システム動作中にいつでも発行することができる。なぜなら、図７のプロセスは、任意のゾーンが再配置、追加、または削除された後にシステム動作を続行するのに必要なアドレス調整を扱うからである。

【０１０３】次にステップ９２に進むと、システム管理者（人間）、またはホスト・プログラム、または論理区画中の割り当てられたゲスト・プログラムが、既にプログラム可能メモリ中で置換ＺＩＴ（replacement ZIP）をセットアップしている。置換ＺＩＴは、システム内の任意のゾーンについて未変更のゾーン原点またはゾーン限界値を含み、任意のゾーン原点値またはその限界値を変更し、１つまたは複数のゾーン・エントリを新規ゾーンとして追加し、あるいは１つまたは複数のゾーン・エントリを削除してシステムからそのゾーンを削除することができる。現在活動状態の（置き換えられる）ＺＩＴは、（プログラム・アドレスではアクセスできない）「システム・エリア記憶域」にある。

【０１０４】次にステップ９３で、新規ＺＩＴの置換および活動化を可能にするために、特殊コマンドによる動作で、システム内のすべてのＣＰＵによる実行が一時的に中断される。ＺＩＴが置き換えられている間に、ＣＰＵがこのＺＩＴにアクセスできるようにすることはできない。そのようにすると、変更プロセス中にエラーが発生する可能性があるからである。

【０１０５】新規ＺＩＴの活動化はステップ９４で行われる。活動化プロセッサは、システム内のＣＰＵの１つ、または特殊プロセッサ（たとえば、サービス・プロセッサまたは何らかの補助プロセッサ）とすることができる。活動化プロセスは、特殊コマンドのハードウェア／マイクロコードにより実行される。すなわち、置換ＺＩＴを、メモリ内のプログラム可能領域から、ＷＲ命令ハードウェア／マイクロコードが活動状態のＺＩＴにアクセスするシステム・エリア記憶域内の位置に存在する古いＺＩＴ上にコピーすることによって実行される。

【０１０６】ステップ９５で、特殊コマンドが、新たに活動化されたＺＩＴの内容とＣＰＵウィンドウ指定テーブルの内容を使用して、各ＣＰＵ内のＷＡＲおよびＷＲの内容について必要な検査および再計算を実行する。ＺＩＴ中でゾーンが明示的に指定されている場合、ＷＡＲの内容がどの新規限界をも超えないことを確認するために、各ＷＡＲ中の値がホスト・ゾーン限界と突き合わせて検査される。新規限界を超えた場合、関連するＷＡＲの内容が無効にされ（かつ、そのＷＡＲによって定義されるウィンドウと関連するＷＲも無効にされ）、ホストに、ＷＡＲ中でアドレス指定されたウィンドウＰＦの位置を変更するためにＷＡＲロード命令を再発行するよう要求する、例外信号が生成される。限界検査によって、ホスト・ゾーン限界を超えていないと判定された場合、

ＷＡＲ中のスモール・アドレスが有効になる。

【０１０７】各ＷＡＲの内容が検査された後、各ＣＰＵウィンドウ指定テーブル・エントリ（図８を参照）中で表された各ＷＲの内容を検査し再計算することによって、ステップ９５のプロセスが続行する。各ＣＰＵウィンドウ指定テーブル・エントリがアクセスされ、それに含まれるゲスト・ゾーン識別子およびゲストのスモール・アドレスが再計算に使用される。ゾーン識別子は、ゲスト・ゾーンのＺＩＴエントリにアクセスするのに使用される。その際、限界を超えていないかどうか検査するために、ゾーンの限界がゲストのスモール・アドレスと比較される。限界を超えている場合、ゲスト・スモール・アドレスが無効化され、エントリ中のラージ・アドレスが無効化される（たとえば、ラージ・アドレスがゼロに設定され、あるいは有効ビットがオフに設定される）。ゲストのスモール・アドレスが限界内である場合、図３でウィンドウ・エントリに関して示したウィンドウ活動化命令プロセスを使用することによって、テーブル・エントリ中の対応するラージ絶対アドレスが再計算され、再計算された値が、対応するＷＲおよびＣＰＵウィンドウ指定テーブル・エントリに入れられる。ＣＰＵウィンドウ指定テーブルのいずれかのゾーンが新規ＺＩＴ中にない場合、そのゾーンは削除されている。その場合、スモール・アドレス・フィールドとラージ・アドレス・フィールドはどちらも、ＣＰＵウィンドウ指定テーブル中のそのウィンドウ・エントリについて無効に設定され、関連するＷＲエントリの内容が無効に設定される。

【０１０８】次にステップ９６で、システム内のすべてのＣＰＵが、先に中断された場所から続行することによってその通常の命令処理を再開する。次に、特殊命令動作が完了し、ステップ９７で終了する。

【０１０９】ＣＰＵウィンドウ指定テーブル－図８

図８は、ＣＰＵウィンドウ指定テーブルを表している。（ゲストおよびホストを実行することが可能な）システム内の各ＣＰＵは、それ自体のウィンドウ指定テーブルがシステム・エリア記憶域に格納されている。

【０１１０】ＣＰＵウィンドウ指定テーブルは、それぞれのＣＰＵに組み込まれた各ウィンドウ用のそれぞれのエントリを含む。テーブル中のウィンドウ・エントリは、ウィンドウ番号のＷ０およびＷ１で索引付けされる。各エントリは、それぞれのウィンドウに現在割り当てられているゾーン番号、関連するＷＲに現在含まれているラージ絶対アドレス、および同じエントリ中のラージ絶対アドレスによって表されるスモール・アドレスを含む。エントリに現在割り当てられているウィンドウがない場合、エントリにゼロが格納される（あるいは、オプションの有効ビットがオフ状態に設定される）。

【０１１１】エントリ中の各フィールドは、本明細書の他の節で説明したとおりに処理される。

【図面の簡単な説明】

【図１】本発明の実施例を使用するコンピュータ・システム内にＮ個のゲスト・ゾーンを有するシステム・メモリとシステム・エリア記憶域の簡略化したマップである。

【図２】ウィンドウ・アクセス・レジスタ（ＷＡＲ）をロードするための「ＷＡＲロード」命令によって使用されるハードウェアおよびプロセスを表す図である。

【図３】大型メモリの（任意のゾーンにおける）任意の位置にアクセスすることを目的として、任意のウィンドウ・レジスタ（ＷＲ）にラージ・アドレスをロードするために「ウィンドウ活動化」命令によって使用されるハードウェアおよびプロセスを表す図である。

【図４】ゲスト・ラージ・アドレスを出力し、このアドレスでホストのオペランド・スモール・アドレスを置き換えるためにＷＲを選択する、好ましい実施例で使用されるプロセスを表す図である。

【図５】ホスト・ウィンドウＰＦに等しいホスト・オペランド・スモール・アドレスをラージ・アドレスで置き換えるためにＷＲを選択する別のプロセスを表す図である。

【図６】大型メモリ内の任意の位置にアクセスするためにラージ・アドレスを生成する際にスモール・ホスト・アドレスを使用し、アドレス置換によって任意のゲスト・ゾーン位置にアクセスするプロセスのフロー・チャートである。

【図７】システムのメモリ内のゲスト・ゾーンおよびホスト・ゾーンを再構成するために、大型メイン・メモリ内の各ゾーンを動的に再配置（変更）するプロセスのフロー・チャートである。

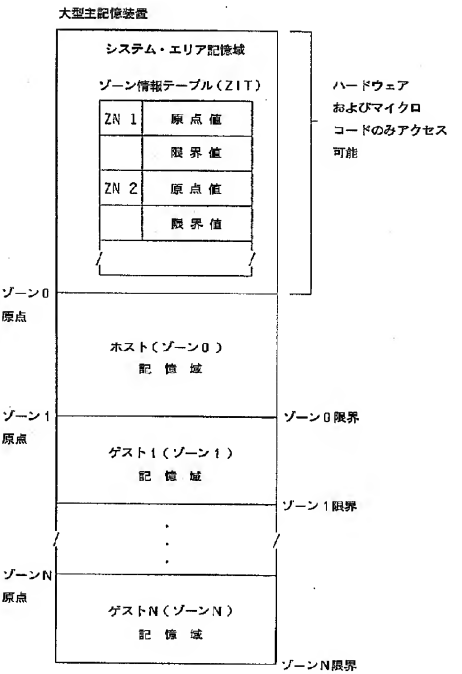
【図８】動的ゾーン再配置プロセスで使用されるＣＰＵウィンドウ指定テーブルを表す図である。

【図９】ホストによるアクセスを必要とするゲスト・ゾーン内の位置をホストが決定するためのホスト・テーブルにおける状態記述（ＳＤ）エントリの例を示す簡略化した図である。

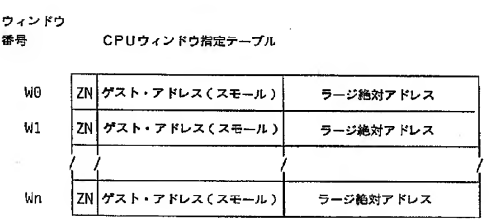
【符号の説明】

４２　Ｒ１汎用レジスタ
４３　Ｒ３汎用レジスタ
４４　ゾーン情報テーブル（ＺＩＴ）
４５　比較機構
４７　ＣＰＵウィンドウ指定テーブル・エントリ
４８　ウィンドウ・レジスタ（ＷＲ）
４９　加算機構
５４１　入力レジスタ
５４２　比較機構
５４４　プレフィクス・レジスタ
５４５　ウィンドウ・アクセス・レジスタ（ＷＡＲ）
５５０　選択機構
５５１　出力レジスタ

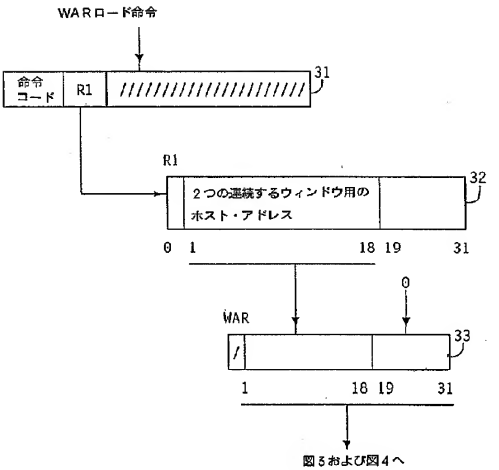
【図１】


大型主記憶装置
システム・エリア記憶域
ゾーン情報テーブル（ＺＩＴ）
ＺＮ １
原点値
限界値
ＺＮ ２
原点値
限界値
ハードウェア
およびマイクロ
コードのみアクセス
可能
ゾーン０
原点
ホスト（ゾーン０）
記憶域
ゾーン１
原点
ゾーン０限界
ゲスト１（ゾーン１）
記憶域
ゾーン１限界
ゾーンＮ
原点
ゲストＮ（ゾーンＮ）
記憶域
ゾーンＮ限界

【図８】

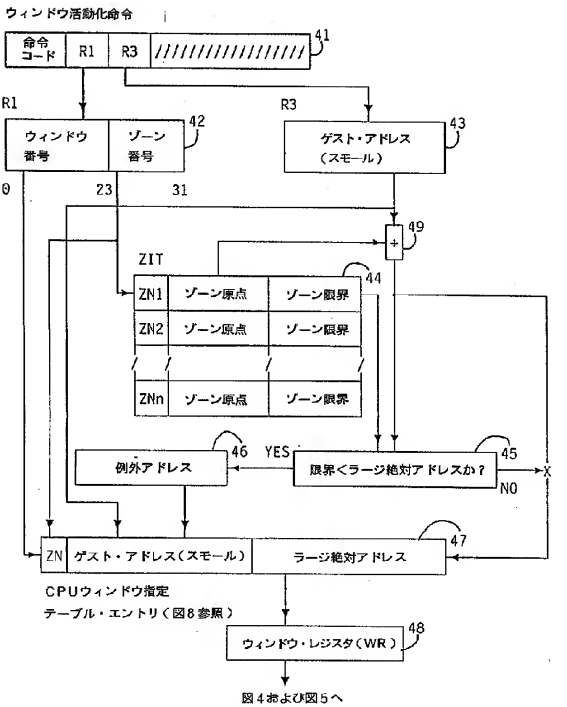

ウィンドウ
番号
ＣＰＵウィンドウ指定テーブル
Ｗ０
ＺＮ
ゲスト・アドレス（スモール）
ラージ絶対アドレス
Ｗ１
ＺＮ
ゲスト・アドレス（スモール）
ラージ絶対アドレス
Ｗｎ
ＺＮ
ゲスト・アドレス（スモール）
ラージ絶対アドレス

【図２】

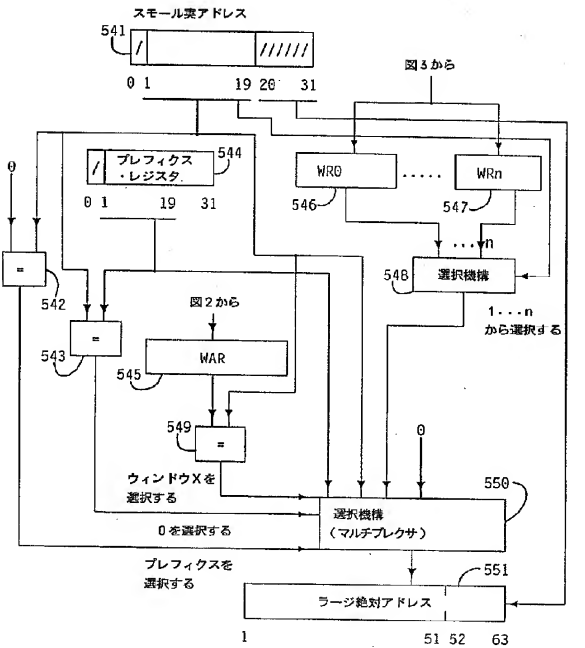


指定された２つの連続するＰＦはそれぞれWR０およびWR１と関連付けられる

【図３】


ウィンドウ活動化命令
命令コード
R1
R3
41
R1
R3
ウィンドウ番号
ゾーン番号
42
ゲスト・アドレス（スモール）
43
0
23
31
49
ZIT
44
ZN1
ゾーン原点
ゾーン限界
ZN2
ゾーン原点
ゾーン限界
ZNn
ゾーン原点
ゾーン限界
例外アドレス
46
YES
限界くラージ絶対アドレスか？
45
X
NO
ZN
ゲスト・アドレス（スモール）
ラージ絶対アドレス
47
CPUウィンドウ指定
テーブル・エントリ（図８参照）
ウィンドウ・レジスタ（WR）
48
図４および図５へ

【図４】


スモール実アドレス
541
0 1
19 20
31
図３から
プレフィクス
・レジスタ
544
WR0
WRn
546
547
548
選択機構
1...n
から選択する
542
543
図２から
WAR
545
549
0
ウィンドウXを
選択する
550
選択機構
（マルチプレクサ）
0を選択する
プレフィクスを
選択する
551
ラージ絶対アドレス
1
51 52
63

【図５】

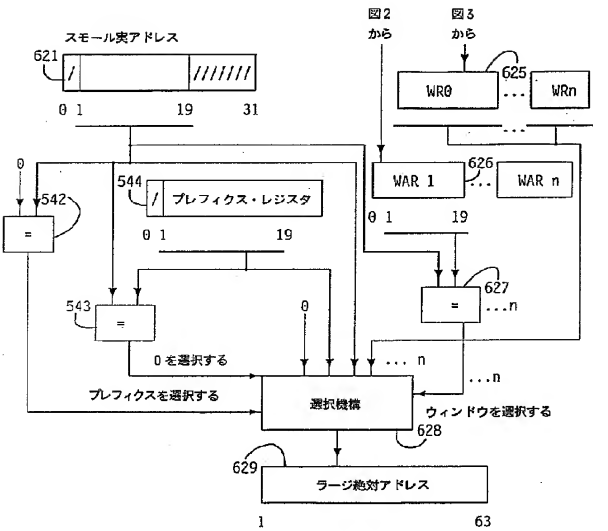
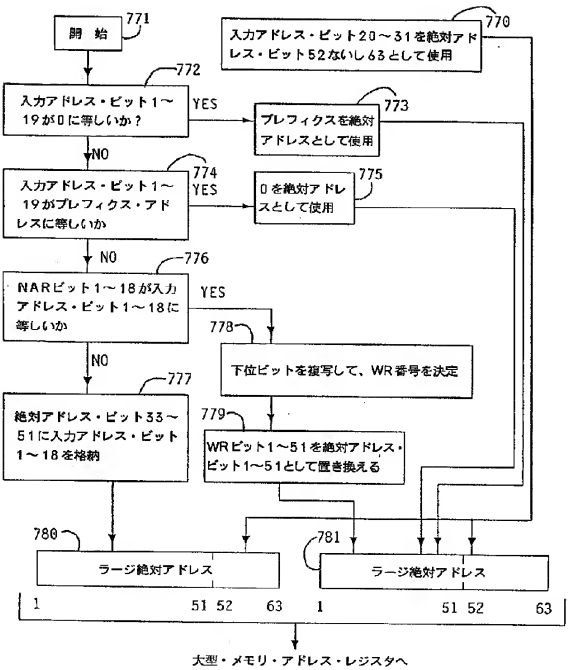
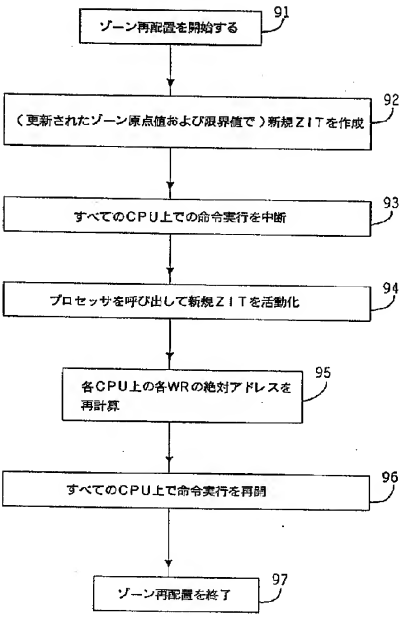


図2
から
図3
から
スモール実アドレス
621
0 1
19
31
625
WR0
WRn
626
WAR 1
WAR n
0 1
19
0
542
544
プレフィクス・レジスタ
0 1
19
627
...n
543
0
0を選択する
... n
...n
プレフィクスを選択する
選択機構
ウィンドウを選択する
628
629
ラージ絶対アドレス
1
63

【図６】

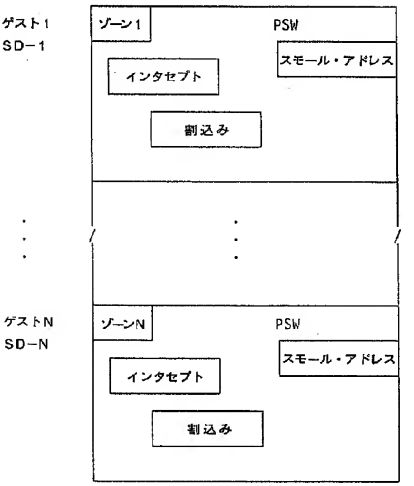

開　始
771
入力アドレス・ビット20～31を絶対アドレス・ビット52ないし63として使用
770
772
入力アドレス・ビット1～19が0に等しいか？
YES
プレフィクスを絶対アドレスとして使用
773
NO
774
入力アドレス・ビット1～19がプレフィクス・アドレスに等しいか
YES
0を絶対アドレスとして使用
775
NO
776
NARビット1～18が入力アドレス・ビット1～18に等しいか
YES
778
NO
下位ビットを複写して、WR番号を決定
777
絶対アドレス・ビット33～51に入力アドレス・ビット1～18を格納
779
WRビット1～51を絶対アドレス・ビット1～51として置き換える
780
781
ラージ絶対アドレス
ラージ絶対アドレス
1
51 52
63
1
51 52
63
大型・メモリ・アドレス・レジスタへ

【図７】


ゾーン再配置を開始する
91
（更新されたゾーン原点値および限界値で）新規ＺＩＴを作成
92
すべてのＣＰＵ上での命令実行を中断
93
プロセッサを呼び出して新規ＺＩＴを活動化
94
各ＣＰＵ上の各ＷＲの絶対アドレスを再計算
95
すべてのＣＰＵ上で命令実行を再開
96
ゾーン再配置を終了
97

【図９】

（ホスト・ゾーン中の）ＳＤテーブル



フロントページの続き

(72)発明者　ロジャー・エルドレッド・フー
アメリカ合衆国12528、ニューヨーク州ハイランド、シャロン・ドライブ　7
(72)発明者　モーン・ジュ・キム
アメリカ合衆国12590、ニューヨーク州ワッピンガーズ・フォールス、シャーウッド・ロード　26
(72)発明者　アレン・ヘルマン・プレストン
アメリカ合衆国12601、ニューヨーク州ポーキープシー、イースト・シー・ストリート　97
(72)発明者　デーヴィッド・エメット・ストゥッキ
アメリカ合衆国12603、ニューヨーク州ポーキープシー、フォックス・ラン　123
(72)発明者　チャールズ・フランクリン・ウェッブ
アメリカ合衆国12603、ニューヨーク州ポーキープシー、マイネッティ・ドライブ　4